

明 細 書

工程シミュレーションシステム

技術分野

- [0001] 本発明は、複数の構成要素(例えば、加工ユニットやクランプユニット、搬送ユニット等)が相互に連繋して動作する生産システム(例えば、トランスファライン等)の工程シミュレーションを行なう工程シミュレーションシステムに関するものである。

背景技術

- [0002] 従来、このような工程シミュレーションシステムとして、特許文献1(特開平4-64164号公報)には、生産システムの各構成要素のシミュレーションモデルを表形式で表現し、これらのモデルを内部記憶部に展開して、関連するモデルの間で関連フレームリストを参照しつつデータのやり取りを行ないながらシミュレーションを行なうものが記載されている。なお、各構成要素のシミュレーションモデルとしては、オペレータや生産セル、搬送系に関するモデルが例示されている。
- [0003] 特許文献2(特開昭61-61752号公報)には、工作機械や段取りステーション、搬送装置、加工ワーク等に関する変数データを入力データファイルに入力し、その変数データに基づいて工程シミュレーションを行ない、シミュレーション結果を出力するものが記載されている。
- [0004] 特許文献3(特開平10-335193号公報)には、LSI製造工程の装置の夫々について、シミュレーションモデル及びパラメータを別個に設定することができるものが記載されている。

発明の開示

- [0005] 本発明はこのような背景技術に鑑みてなされたものであり、生産システムの工程シミュレーションを行なうためのシミュレーションプログラムを容易に生成することができ、また、シミュレーション対象となる生産システムにおける工程の増加や削除等にも容易に対応することができる、工程シミュレーションシステムを提供することを目的とする。
- [0006] 本発明は、生産システムの工程シミュレーションを行なう工程シミュレーションシステ

ムにおいて、生産システムを構成する各種の構成要素の配置を工程識別名称と構成要素名称との組み合わせによりワークの流れに沿って記述してなる要素配置データから、構成要素名称を工程識別名称と組み合わせた状態で順次読み込む要素配置データ読込手段と、前記各種の構成要素の動作シミュレーションを行なうためのシミュレーションプログラムを前記各種の構成要素ごとに記述してなる複数の要素定義ファイルの中から、前記要素配置データ読込手段により読み込まれた構成要素名称に対応する要素定義ファイルを順次読み込む要素定義ファイル読込手段と、前記要素定義ファイル読込手段により読み込まれた前記各要素定義ファイルに記述されているシミュレーションプログラムを順に配列してシミュレーションプログラム配列を作成するプログラム配列作成手段と、前記プログラム配列作成手段により作成された前記シミュレーションプログラム配列に含まれる一連のシミュレーションプログラムを実行して、前記生産システムを構成する前記各種の構成要素の動作をシミュレートするプログラム実行手段とを備えたことを特徴とする工程シミュレーションシステムを提供する。

[0007] なお、本発明において、前記要素配置データは、文字情報の追加や削除等の編集が可能なソフトウェアを用いて作成される表形式のデータからなり、列方向にワークの流れ方向が設定され、行方向に工程識別名称と構成要素名称とが記述されていることが好ましい。

[0008] また、本発明において、前記各要素定義ファイルは、自己のシミュレーションプログラムを記述したプログラム記述部と、自己のシミュレーションプログラムで使用される変数を記述した変数記述部とを含み、前記変数記述部には、他の要素定義ファイルの変数を参照する要素定義ファイルであれば、参照先変数を指定する外部参照変数が定義され、他の要素定義ファイルに変数を参照させる要素定義ファイルであれば、外部参照変数により参照される取出変数が定義されており、前記要素定義ファイル読込手段により読み込まれた前記各要素定義ファイルの前記変数記述部に記述されている全ての変数を含む変数配列を作成する変数配列作成手段と、前記変数配列作成手段により作成された前記変数配列に含まれる前記外部参照変数と前記取出変数とを対応付ける変数対応手段とをさらに備えることが好ましい。

[0009] さらに、本発明において、前記要素配置データには、前記各要素定義ファイルの前

記変数記述部に記述されている変数名を別の変数名に置換するための変数名置換データが記述されており、前記要素配置データにおいて前記変数名置換データが設定されている変数名を前記変数名置換データに記述されている別の変数名に置換する変数名置換手段をさらに備えることが好ましい。

[0010] さらに、本発明において、前記変数配列作成手段により作成された前記変数配列には、複数の構成要素間に亘る複数の同種の変数をまとめて管理するための登録エリアを所要数備えた変数テーブルが作成されており、前記各要素定義ファイルのうち、まとめて管理される同種の変数を含んでいる要素定義ファイルには、当該変数に応じて作成される前記変数配列内の対応する変数テーブルに、前記変数配列内における当該変数の変数位置を登録する初期処理プログラムが記述されており、前記プログラム配列作成手段は、前記要素定義ファイル読込手段により読み込まれた前記各要素定義ファイルに記述されている初期処理プログラムを順に配列して初期処理プログラム配列を作成し、前記プログラム実行部は、前記プログラム配列作成手段により作成された前記初期処理プログラム配列に含まれる初期処理プログラムを実行して、まとめて管理される同種の変数について、前記変数配列内の変数テーブルに、前記変数配列内における当該変数の変数位置を登録することが好ましい。

[0011] さらに、本発明において、前記各要素定義ファイルの前記プログラム記述部に記述されているシミュレーションプログラムは、ラダー言語方式の命令群で記述されていることが好ましい。

[0012] なお、本発明に係る工程シミュレーションシステムは、当該工程シミュレーションシステムとしてコンピュータを機能させるための工程シミュレーションシステム用プログラムを記録したコンピュータ読み取り可能な記録媒体の形態で提供されてもよい。

[0013] 本発明によれば、生産システムを構成する各種の構成要素の配置を工程識別名称と構成要素名称との組み合わせによりワークの流れに沿って記述した要素配置データを準備しておく。そして、このようにして準備された要素配置データに従って工程識別名称とともに構成要素名称を順次読み込むとともに、この読み込まれた構成要素名称に対応する要素定義ファイルを順次読み込んでそこに記述されているシミュレーションプログラムを順にシミュレーションプログラム配列に格納し、最終的に実行さ

れる一連のシミュレーションプログラムを作成する。このため、要素配置データ上で異なる配置位置に同一の構成要素が存在している場合でも、これらの構成要素は、構成要素名称と工程識別名称との組み合わせにより別の構成要素として識別されることとなる。従って、同一の構成要素の要素定義ファイルについては、1種類のみを記述しておけば足り、シミュレーションプログラムを作成するにあたっての要素定義ファイルの作成作業が容易になる。また、要素配置データに基づいてシミュレーションプログラム配列が作成されるので、別の工程シミュレーションを行なうために要素配置データが変更されたときでも、その変更された要素配置データに対応する一連のシミュレーションプログラムを容易に作成することができる。

[0014] また、本発明において、文字情報の追加や削除等の編集が可能なソフトウェアを用いて要素配置データを作成するようにすれば、要素配置データに新たに工程識別名称と構成要素名称との組み合わせを追加したり削除したりすることがソフトウェアの編集機能を利用して容易に行えるようになり、シミュレーション対象となる生産システムにおける工程の増加や削除等を反映した要素配置データを極めて容易に作成することができる。

[0015] さらに、本発明において、要素定義ファイル内に記述されている変数名を、要素配置データに記述された変数名置換データに基づいて別の変数名に置換するようにすれば、例えば、要素定義ファイルにおいて一般的な名称で記述された外部参照変数の参照先変数の名称を、要素配置データに記述された具体的な名称に置換することにより、1つの要素定義ファイルを用いて種々の参照先を指定することができる。また、例えば、同一の工程において同種の加工ユニットが複数配置されるときには、加工ユニットを示す一般的な変数名を加工ユニットを区別する別の変数名に置換することにより、1つの要素定義ファイルを用いて異なる加工ユニットを区別することができる。

[0016] さらに、本発明において、まとめて管理すると都合のよい変数(例えば、機械の故障状態といった変数)について、要素定義ファイルに予めその旨を記述しておき、それらの変数を、予め準備した変数テーブルでまとめて管理するようにすれば、機械の故障状態を探して作業員が修理に向かう、といったシミュレーションを行なうときでも、そ

のまとまった変数テーブルを用いて変数状態の判別を行なうことにより機械の故障状態を探ることが可能となるので、生産システムの工程シミュレーションを短い実行時間で行なうことができる。

- [0017] さらに、本発明において、シミュレーションプログラムをラダー言語方式で記述するようにすれば、複雑な分岐命令を用いることなく構成要素の動作シミュレーションを記述することが可能となり、シミュレーションプログラムを含む要素定義ファイルを容易に作成することができる。

図面の簡単な説明

- [0018] [図1]は、本発明の一実施の形態に係る工程シミュレーションシステムの全体構成を示すブロック図である。
- [図2]は、図1に示す工程シミュレーションシステムで用いられる要素配置データの一例を示す図である。
- [図3]は、図1に示す工程シミュレーションシステムで用いられる要素定義ファイルの一般的な構成を示す図である。
- [図4]は、図3に示す要素定義ファイルの具体例を示す図である。
- [図5]は、図3に示す要素定義ファイルの具体例を示す図である。
- [図6]は、図1に示す工程シミュレーションシステムで実行されるシミュレーションプログラムに含まれる命令群の具体例を示す図である。
- [図7]は、図1に示す工程シミュレーションシステムの動作を説明するためのフローチャートである。
- [図8]は、図1に示す工程シミュレーションシステムの動作過程で生成される要素配列の一例を示す図である。
- [図9]は、図1に示す工程シミュレーションシステムの動作過程で生成される変数配列の一例を示す図である。
- [図10]は、図1に示す工程シミュレーションシステムの動作過程で生成される変数配列の一例を示す図である。
- [図11]は、図1に示す工程シミュレーションシステムの動作過程で生成される変数配列における外部参照変数と取出変数との対応関係を説明するための図である。

[図12]は、図1に示す工程シミュレーションシステムの動作過程で生成される変数配列における外部参照変数と変数テーブルとの対応関係を説明するための図である。

[図13]は、図1に示す工程シミュレーションシステムの動作過程で生成される外部参照変数対応配列の一例を示す図である。

[図14]は、図1に示す工程シミュレーションシステムの動作過程で生成される初期処理プログラム配列の一例を示す図である。

[図15]は、図1に示す工程シミュレーションシステムの動作過程で生成されるシミュレーションプログラム配列の一例を示す図である。

[図16]は、図1に示す工程シミュレーションシステムで用いられる要素配置データの他の例(変数名置換データの他の適用例)を示す図である。

[図17]図17A及び図17Bは、シミュレーション対象となる生産システムの一例を示す工場配置図である。

発明を実施するための形態

[0019] 以下、図面を参照して本発明の実施の形態について説明する。

先ず、図1により、本発明の一実施の形態に係る工程シミュレーションシステムの全体構成について説明する。

[0020] 図1に示すように、工程シミュレーションシステム1は、生産システムの工程シミュレーションを行なうためのものであり、シミュレーションシステム本体1aと、シミュレーションシステム本体1aに接続された入力装置11及び出力装置12とを備えている。なお、入力装置11は、シミュレーションシステム本体1aに対して各種の指示(ソフトウェアの起動や停止等)を入力するためのものであり、マウスやキーボード等からなっている。また、出力装置12は、シミュレーション結果等を出力(表示)するためのものであり、CRT等からなっている。

[0021] このうち、シミュレーションシステム本体1aは、要素配置データ3と要素定義ファイル4とに基づいてシミュレーションプログラムを作成するプログラム作成部2と、プログラム作成部2により作成されたシミュレーションプログラムを実行するプログラム実行部10とを備えている。

[0022] プログラム作成部2は、要素配置データ3と要素定義ファイル4とに基づいて、要素

配列5、変数配列6、外部参照変数対応配列7、初期処理プログラム配列8及びシミュレーションプログラム配列9を作成するものであり、要素配置データ読込部2a、要素定義ファイル読込部2b、初期処理プログラム配列作成部2c、シミュレーションプログラム配列作成部2d、変数配列作成部2e、変数対応部2f及び変数名置換部2gを有している。なお、これらの各部2a〜2gからなるプログラム作成部2の詳細については後述する。

[0023] プログラム実行部10は、プログラム作成部2により作成された初期処理プログラム配列8に含まれる初期処理プログラムを実行して初期処理を行なうとともに、シミュレーションプログラム配列9に含まれる一連のシミュレーションプログラムを実行して、生産システムを構成する各種の構成要素の動作をシミュレートするものである。なお、プログラム実行部10の詳細については後述する。

[0024] なお、シミュレーションシステム本体1aのプログラム作成部2及びプログラム実行部10は、シミュレーション目的のためのコンピュータプログラム(工程シミュレーションシステム用プログラム)をパーソナルコンピュータ(PC)上で稼働させることにより実現することが可能である。なお、このような工程シミュレーションシステム用プログラムは、メモリやハードディスク、フレキシブルディスク、CD-ROM、DVD等のようなコンピュータ読み取り可能な記録媒体に格納され、コンピュータのプロセッサから逐次読み出されて実行されることによりプログラム作成部2及びプログラム実行部10の機能を実現する。

[0025] 以下、図1に示す工程シミュレーションシステム1の詳細を具体的に説明するため、工程シミュレーションが行なわれる具体的な生産システムとして、前ストッカにストックされたワークをコンベアで接続台に送り込み、その接続台に送り込まれたワークを、リフトアンドキャリー方式の搬送装置で搬送しながら2つの加工ユニットで加工して、排出側の接続台に排出し、その排出側の接続台からコンベアにより搬出して後ストッカにストックする、という生産システムを例に挙げる。

[0026] 図17A及び図17Bに示すように、この生産システムにおいては、前ストッカ101にワークWがストックされており、前ストッカ101のプッシャ101aによりコンベア102にワークWが送り込まれる。

[0027] ここで、接続台103は、ワーク搬送方向の両側に載置部分103a, 103aを有している。これらの載置部分103a, 103aの間にはコンベア102が入り込んでいて、コンベア102が周回することによりワークWが接続台103上に送り込まれる。また、接続台109は、ワーク搬送方向の両側に載置部分109a, 109aを有している。これらの載置部分109a, 109aの間にはコンベア110が入り込んでいて、コンベア110が周回することにより接続台109からワークWが取り出されて後ストック111へ受け渡される。

[0028] リフトアンドキャリー (LF搬送) 装置104は、前後の接続台103, 109と2つの加工ユニット(加工機械) 106, 106とを繋いでいる。加工ユニット106, 106は夫々回転工具107を有しており、工具軸線方向に関して原位置と加工位置との間で進退する。加工ユニット106, 106の前方には、ワーク搬送方向の両側に設けられたワーク載置部108, 108が配置されている。なお、ワーク載置部108, 108に関連してワーク固定用の治具が設けられているが、ここでは図示を省略している。

[0029] 以上のようにしてワーク搬送方向の両側に位置する接続台103, 109の載置部分103a, 109a及びワーク載置部108, 108の間には、リフトアンドキャリー装置104の搬送部材105が配置されている。リフトアンドキャリー装置104の搬送部材105は、待機位置Aから所定高さだけ上昇して載置部分103a及びワーク載置部108に載置されているワークWを持ち上げるとともに(上昇位置B)、接続台103、各加工ユニット106及び接続台109の間の搬送方向配置ピッチPだけ搬送方向に前進し(上昇前進位置C)、その後、下降前進位置Dまで下降する。これにより、接続台103に位置付けられていたワークWは搬送方向前方の加工ユニット106へ受け渡され、搬送方向前方のユニット106に位置付けられていたワークWは搬送方向後方の加工ユニット106へ受け渡され、搬送方向後方の加工ユニット106に位置付けられていたワークWは接続台109へ受け渡される。

なお、この生産システムには、機械保全のための人員が2名(保全メンバA, B)居るものとする。

[0030] 以上に説明したような生産システムの工程シミュレーションを行なうときには、図1に示す工程シミュレーションシステム1では、先ず、シミュレーション対象となる生産システムを構成する構成要素(装置)の配置を記述した要素配置データ3を準備する。要

要素配置データ3は、生産システムを構成する各種の構成要素の配置を工程識別名称と構成要素名称との組み合わせによりワークの流れに沿って記述したものであり、基本的には工場内の装置配置図に対応している。なお、要素配置データ3は、文字情報の追加や削除、置換等の編集を行なうことのできるソフトウェア（例えば、汎用の表計算ソフト(MICROSOFT社のEXCEL(登録商標))）で作成される表形式のデータ(データシート)である。このような表形式のデータシートは、テキストエディタのような編集ソフトを用いて作成することもできる。

[0031] 表形式のデータシートである要素配置データ3は、基本的には、図2に示すように、列方向(表の縦方向:行並び方向)にワークの流れ方向が設定され、行方向(表の横方向:列並び方向)に、工程識別名称と構成要素名称とが、工程識別名称で区別される工程ごとに1行ずつ記述されている。具体的には、各行のB列セルにはモジュール名が、C列セルには工程名が、D列セル以後の任意のセルには構成要素名称が記述されている。工程識別名称は、同一行のモジュール名と工程名とからなり、工程名が省略されている場合もある。1つの要素配置データ3内で、工程識別名称は全て互いに異なる名称となるように記述されている。

[0032] 要素配置データ3では、同一の仕様の構成要素は同一の名称で記述されている。ここでは、2つの加工ユニットがすべて同一の“ユニット1”であり、2つのジグが全て同一の“ジグ1”であり、さらに2つのコンベア、接続台、保全メンバも夫々同一の仕様であるので、全て同一の名称で記述されている。

[0033] 要素配置データ3には、実際の工場には配置されることのない構成要素名称も記述されている。構成要素“総合1”は、モジュール140T内で、生産システムの工程シミュレーションの実行時にまとめて管理するのが便利である変数情報を集めておくための要素であり、後述するように、2つのジグ1のアンクランプに関する変数をまとめて管理する“アンクランプT”テーブルや、各ユニット1の“原位置”データ(図示せず)をまとめて管理する変数テーブルを持ち、LF搬送に対して、そのアンクランプ情報に基づいて、全てのジグがアンクランプされ、かつ、全てのユニットが原位置にあるときに、搬送可能という情報を流す、というシミュレーションを行なう。また、構成要素“保全指令1”は、生産システムの工程シミュレーションの実行時にまとめて管理するのが

便利である変数情報を集めておくための要素であり、具体的には後述するが、例えば、全ての機械の稼動状況や保全メンバの作業状況等のデータを集めておく。

[0034] 要素配置データ3に記述されている、これらの構成要素名称の夫々に対応して、図3にその一般的な構成を示すような要素定義ファイル4が準備されている。なお、図4及び図5は、要素配置データ3に記述された構成要素名称の夫々に対応する具体的な要素定義ファイル4a〜4jを示すものである。

[0035] 図3に示すように、要素定義ファイル4は、その一般的な構成として、先頭に要素名称記述部4Aを含み、続いて変数記述部4B及びプログラム記述部4Cを含む。

[0036] このうち、変数記述部4Bには、自己のシミュレーションプログラムで使用される内部変数(例えば、ユニット1(加工ユニット)では、加工時の前進速度や後退速度等)が初期値と共に記述される。また、変数記述部4Bには、他の要素定義ファイルの変数を参照する要素定義ファイルであれば、参照先変数を指定する外部参照変数が定義され、他の要素定義ファイルに変数を参照させる要素定義ファイルであれば、外部参照変数により参照される取出変数が初期値と共に定義される。

以下、外部参照変数及び取出変数について具体的に説明する。

[0037] 外部参照変数は、次のルールに従って記述されている。

“自ファイルでの変数名(外部参照変数) 参照先変数名(又は参照先変数テーブル名) 参照先工程名 モジュール名”

ここで、参照先工程名及びモジュール名は、その外部参照変数が定義される要素定義ファイルの構成要素に対して、相対的な位置関係を示す語を用いて記述されてもよいし、特定のモジュール名及び工程名を記述してもよい。なお、“相対的な位置関係を示す語”とは、“前工程”、“自工程”、“次工程”、あるいは、“前モジュール”、“自モジュール”、“次モジュール”といった記述である。ここで、“前工程”とは、自要素に対して要素配列データ3上で同一列の上側にある構成要素のモジュール名と工程名とを指し、“自工程”とは、自要素と同一行の構成要素のモジュール名と工程名とを指し、“後工程”は、自要素に対して同一列の下側にある構成要素のモジュール名と工程名とを指す。同様に、“前モジュール”、“自モジュール”、“次モジュール”についても、自モジュールに対する相対的な位置関係を示している。なお、参照先変数名(

又は参照先変数テーブル名)は、指定されたモジュール又は工程内で他の変数名と重複しないようにされている。

- [0038] 具体的には例えば、ユニット1及びジグ1では、同一の工程に含まれるジグ1のクランプ完了をもってユニット1が加工を開始し、加工が完了するとその加工完了信号をもってジグ1がアンクランプする、というシーケンス動作が行なわれる。
- [0039] この場合、このような情報のやり取りをユニット1とジグ1との間で行なうために、“ユニット1”の要素定義ファイル(図5に示す要素定義ファイル4g)では、“自工程加工準備完了”(外部参照変数)として“自工程”の“加工準備完了”(参照先変数名)を参照する、と定義されている。これに対して、“ジグ1”の要素定義ファイル(図5に示す要素定義ファイル4f)では、取出変数として“加工準備完了”(ワークのクランプが完了したことを意味する)が定義されている。また、“ジグ1”の要素定義ファイル(図5に示す要素定義ファイル4f)では、外部参照変数としての“ユニット加工完了”を“自工程”の“加工完了”を参照すると定義し、“ユニット1”の要素定義ファイル(図5に示す要素定義ファイル4g)では、取出変数として“加工完了”が定義されている。従って、これらのジグ1とユニット1とが同一の工程に配置されている状態では、要素配置データ3において“ユニット1”と同一行(すなわち、自工程)に配列された“ジグ1”との間で外部参照変数と取出変数とが互いに対応することになる。他にも同様の外部参照変数及び取出変数の関係が定義されている。例えば、“接続台”の要素定義ファイル(図4に示す要素定義ファイル4c)と“コンベア”の要素定義ファイル(図4に示す要素定義ファイル4b)、“接続台”の要素定義ファイル(図4に示す要素定義ファイル4c)と“LF搬送”の要素定義ファイル(図4に示す要素定義ファイル4e)との間で同様の関係が定義されている。
- [0040] 次に、外部参照変数の参照先変数が変数テーブルである場合について説明する。

図5に示す“ユニット1”の要素定義ファイル4gでは、外部参照変数の一つが、
“保全指令・機械状態 機械状態T NONE 保全指令”
と定義されている。

- [0041] ここで、“NONE”は、工程名が無いことを示している。要するに、変数“保全指令・

機械状態”は、モジュール名“保全指令”の“機械状態T”という変数テーブルに対応する、ということである。その他の構成要素の要素定義ファイルでも、“機械状態”に関する外部参照変数が定義しており、“ジグ1”の要素定義ファイル4fや“LF搬送”の要素定義ファイル4eにも同様の記述が含まれている。

[0042] これに対応するように、図5に示す“保全指令1”の要素定義シート4iには、取出変数として、

[11]機械状態T

と記述されている。

[0043] ここで、括弧([])内の数は装置数である。このような記述があるときには、後述する変数配列6の作成時に、“保全指令1”の要素定義ファイルを読み込んだときの配列エリア中に、“装置数(ここでは11)+1”個のデータ登録エリアを展開して、図12(A)に示すように、“機械状態T[0]ー機械状態T[11]”のテーブル変数を持つ“機械状態T”という変数テーブル20を作成するようになっている。“保全指令1”と“総合1”を除く各構成要素(“保全指令1”と“総合1”は架空の要素であるから、変数“機械状態”はない)の“保全指令・機械状態”という外部参照変数は、この取出変数としてのテーブル変数“機械状態T[1]ーT[11]”に順に対応付けられる。同様の変数テーブル20として、図9及び図10に示す変数配列6には、“保全指令1”に関連して作成される“作業員状態T”テーブルや、“総合1”に関連して作成される“アンクランプT”テーブルが例示されている。

[0044] なお、図2に示すような要素配置データ3において構成要素名称が配置されている結果、外部参照変数の参照先を、単純に自工程の前工程や次工程との関連で記述することができないような場合もある。例えば、図2に示す構成要素“接続台”は、前後の搬送要素を繋ぐものであり、前後の搬送要素を外部参照変数の参照先としている。ここで、図2に示す要素配置データ3では、ワークの流れ方向の上流側の構成要素“接続台”では、前工程は“140T・コンベア”であるが、後工程は“140T・1ST”であって、前後の参照先が整合しない。また、下流側の構成要素“接続台”では、前工程は“140T・2ST”となり、後工程は“140T・搬出コンベア”であって、これも前後の参照先が整合しない。

[0045] そこで、図4に示す“接続台”の要素定義ファイル4cでは、外部参照変数の参照先工程名を“後搬送”や“前搬送”といった一般的な名称で記述する一方で、図2に示す要素配置データ3には、このような一般的な名称で記述された外部参照変数の参照先工程名を別の名称に置換するための変数名置換データを記述しておき、変数配列6を作成する際に置換する。

[0046] すなわち、要素配置データ3において、上流側の構成要素“接続台”では、前搬送が“コンベア”であり、後搬送が“搬送”であり、一方、下流側の構成要素“接続台”では、前搬送が“搬送”であり、後搬送が“搬出コンベア”であり、各構成要素“接続台”に関して前搬送及び後搬送の対象が異なっているので、本来は、別々のコンベアとして要素定義ファイルを作成しなければならない。しかしながら、上述したようにして、変数名を置換するようにすれば、1つの要素定義ファイルでもって、前搬送及び後搬送の工程が異なっている場合にも対応することができる。

[0047] なお、図2に示すように、変数名置換データは、構成要素名称の設定されたセルの次のセルに記述され、その記述は次のルールに従って行なわれる。

“置換後変数名＝置換前変数名”

ここで、複数の変数を置換したいときには、同様の記述が同一行に並べられる。

[0048] なお、上述したような変数名の置換は、外部参照変数の参照先工程名に限らず、各要素定義ファイルに記述されている変数名の全てについて行なうことができる。具体的には例えば、図16に示すように、同一工程“140T・1ST”に2つの同一の加工ユニットが配置されている場合には、単純に“モジュール名＋工程名”の組み合わせでは、夫々を区別することができない。この場合でも、上述したと同様の方法で構成要素名を置換してやれば、夫々を問題なく区別することができる。なお、図16に示す場合には、上側の“ユニット1”が“Lユニット”に置換され、下側の“ユニット1”が“Rユニット”に置換され、また、各ユニットの原位置も夫々置換される。

[0049] 図3に戻って説明を続けると、要素定義ファイル4のプログラム記述部4Cは、初期処理プログラム記述部4C1と、シミュレーションプログラム記述部4C2とを有している。

[0050] このうち、初期処理プログラム記述部4C1には、上述した変数テーブル20に、該当

する変数の変数位置を格納するためのデータ格納命令 (SET TBNO) や、シミュレーションプログラムを実行するための変数についてのデータ設定等を行なうための初期処理プログラムが記述されている。

- [0051] また、シミュレーションプログラム記述部4C2には、夫々の構成要素の動作シミュレーションを行なうためのシミュレーションプログラムが記述されている。
- [0052] 具体的には、構成要素“前ストック”の要素定義ファイルの場合には、後工程となるコンベアからワーク要求信号があつて、前ストック上にワークがあるときに、前ストック上のワークをコンベアに受け渡す、といった動作をシミュレートするためのプログラムが記述されている。
- [0053] 構成要素“コンベア”の要素定義ファイルの場合には、ワークがないときに前工程要素にワーク要求信号を出したり、後工程要素からワーク要求があつて前工程要素からワークが受け渡されているときに前進動作してワークを後工程要素に受け渡す、といった動作をシミュレートするためのプログラムが記述されている。
- [0054] 構成要素“接続台”の要素定義ファイルの場合には、搬送要素と搬送要素とを繋ぐものであつて、ワークの有無を判断して前搬送要素にワーク要求信号を出し、あるいは、次搬送要素にワーク準備完了信号を出す、といった動作をシミュレートするためのプログラムが記述されている。
- [0055] 構成要素“LF搬送”の要素定義ファイルの場合には、前搬送要素からワーク準備完了信号を受け取り、しかも、自モジュール内の全てのジグ1がアンクランプでユニット1も原位置である場合に、リフトアンドキャリー動作を行ない、ワークを順次搬送する、といった動作をシミュレートするためのプログラムが記述されている。
- [0056] 構成要素“ジグ1”の要素定義ファイルの場合には、LF搬送が完了したことを確認してワークをクランプし、ユニット1による加工完了を待ってアンクランプする、といった動作をシミュレートするためのプログラムが記述されている。
- [0057] 構成要素“ユニット1”の要素定義ファイルの場合には、ジグ1によるクランプを確認して原位置から前進し、加工し、後退して原位置に戻る、といった動作をシミュレートするためのプログラムが記述されている。
- [0058] 構成要素“後ストック”の要素定義ファイルの場合には、後ストック上にワークが無け

れば、前工程要素にワークを要求する、といった動作をシミュレートするためのプログラムが記述されている。

[0059] 構成要素“保全メンバ”の要素定義ファイルの場合には、保全を必要とする機械に対して、予め設定してある保全作業時間が経過したかどうかを判断して、保全作業実行中か又は空いているのかを判断する動作をシミュレートするプログラムが記述されている。

[0060] なお、このようにして要素定義ファイル4のプログラム記述部4Cのシミュレーションプログラム記述部4C2に記述されるシミュレーションプログラムは、ストアードプログラムサイクリック処理方式(すなわち、いわゆるラダー言語方式)の命令群で記述されていることが好ましい。

[0061] ここで、構成要素“ユニット1”を例に挙げて、シミュレーションプログラムの具体例について説明する。

[0062] 図6の左側に実際のシミュレーションプログラムが示され、右側にそのシミュレーションプログラムをラダー表示したものが示されている。

[0063] 図6において、“MOTION”命令は、動作のシミュレート命令であり、命令の後に、前進動作、後退動作の別が指示される。この命令が実行される度に、動作対象の位置や速度等の変数の値が刻々変化していく。

“TMR”命令は、タイマのシミュレート命令である。この命令が実行されると、“前進端タイマ設定”という変数に設定された時間が経過して時点で接点が閉じられる。

“MOV”命令は、前の変数の値を、後の変数にコピーする命令である。

“OUT”命令は、後ろの変数をON(“1”)とするものである。

“LD”命令は、回路ブロックの始まりを表す命令である。

“LDF”命令は、判別を伴った回路ブロックの始まりを表す命令である。

“AND”命令は、論理積を表す命令である。

“ANDNOT”命令は、否定の論理積を表す命令である。

“ANDF”命令は、比較を伴った論理積を表す命令である。

“SUB”命令は、先頭の変数から真中の変数の値を差し引いて3番目の変数に入れるという命令である。

“SET”命令は、変数に“1”をセットする命令である。

[0064] 図6に示すシミュレーションプログラムの命令群C1〜C8は、ラダー図であるR1〜R8に夫々対応しているので、両者を対比することにより、上述した各命令の意味が理解されるものと思われる。

[0065] ここで、詳細に説明すれば、命令群C1は、自工程加工準備完了(ジグのクランプが完了していることを意味する)で、かつ、ユニットが原位置にあり、かつ、機械が故障中で無く、かつ、加工が完了していないときに、前進指令を出す、ことを意味している。

[0066] 命令群C2は、前進指令で前進動作させることを意味している。

[0067] 命令群C3は、前進動作の結果、前進端に至ったら前進端タイマをONとすることを意味している。

[0068] 命令群C4は、前進端タイマが設定された時間経過した時点で加工完了とすることを意味している。

[0069] 命令群C5、C6では、加工が完了してユニットが原位置になれば、後退指令を出して後退動作させることを意味している。

[0070] 命令群C7は、故障発生に関するシミュレーションプログラムであり、“故障発生時間count”という変数が“0”以上で、かつ、故障発生実行という変数が“0”以上(これは故障を発生させるという意味である)であることを判断して、両方の条件が整えば、“故障発生時間count”に設定されている時間データからシミュレーションプログラムを実行させるときの1スキャン分の時間(スキャンピッチ)を差し引いて“故障発生時間count”へ格納する、というシミュレーションを行なっている。なお、“故障発生時間count”には、各機械ごとに予め定められている故障発生までの時間(故障発生時間)が初期処理プログラムでコピーされているので(図5に示す“ユニット1”の定義ファイル4gにおける初期処理プログラム(MOV)を参照)、ここではシミュレーションの開始からの経過時間を故障発生時間から減じるという計算をしている。

[0071] 命令群C8は、命令群C7での演算結果が“0”以下となった、すなわち、故障発生するだけの時間が経過したことを判断し、そのときに機械状態が“ゼロ”(故障していない)であれば、機械状態という変数を“1”(故障)とする、というシミュレーションを行

なっている。

- [0072] なお、シミュレーションプログラムの命令としてはこの他にもいくつか準備され、各構成要素の動作シミュレーションを行なうのに必要な命令群を構成している。別の命令としては、例えば、変数テーブル20に対して一定の条件で検索を行なう命令や、回転動作をシミュレートするための命令等がある。
- [0073] 上述したようなラダー言語方式の命令群では、ある命令群で記述した条件が整わない場合には、次の命令群を処理する、というようにシーケンシャルに処理が行なわれる。このため、いくつかの条件に応じていくつかの処理を行なわせるような場合であっても、複雑なIF文を用いてプログラム処理の飛び先をいくつか指定するような形式の条件式(複雑な分岐命令)を用いる必要がなく、シミュレーションプログラムをより簡単な記述で扱いやすいものとすることができる。
- [0074] 次に、図7に示すフローチャートに従って、図1に示す工程シミュレーションシステム1の動作について説明する。
- [0075] まず、プログラム作成部2の要素配置データ読込部2aにより、準備された要素配置データ3を1行読み込む(ステップS1)。次いで、要素配置データ読込部2aは、このようにして読み込まれた1行分のモジュール名、工程名及び要素名を順に図8に示すような要素配列5の1行に並べていく(ステップS2)。図8に示す要素配列5において、モジュール名の前には、要素配列5内での各要素の位置を示すアドレスが記録される。なお、このアドレスは、要素配置データ3を1行読むと、それに対応して1ずつ増えていく。また、要素配列5において、要素名の後ろには、後述する変数配列6における各要素に対応する変数の格納位置が、先頭変数位置と変数個数とで表されるエリアにより示される。このとき、要素配置データ3内に、要素名に関する変数名置換データがあれば、プログラム作成部2の変数名置換部2gにより、置換後の変数名で要素名を置換する。
- [0076] 次に、プログラム作成部2の要素定義ファイル読込部2bにより、要素配置データ読込部2aにより読み込まれた構成要素名称に対応する要素定義ファイル4a〜4jを探し出して読み込む(ステップS3)。
- [0077] そして、プログラム作成部2の変数配列作成部2eにより、要素定義ファイル読込部2

bにより読み込まれた要素定義ファイル(要素定義ファイル4a〜4jのいずれか)の変数記述部4Bに記述されている全ての変数名及び値を、変数の属性(内部変数、取出変数及び外部参照変数)と共に読み出し、図9及び図10に示すような変数配列6に並べる(ステップS4)。図9及び図10に示す変数配列6において、属性の前には、変数配列6内での各変数の位置を示すアドレスが順に付されて記録されていく。ここでも、要素配置データ3内に、変数名に関する変数名置換データがある場合には、プログラム作成部2の変数名置換部2gにより、置換すべき変数名が読み込まれた変数名にあるかを検索して、もし置換すべき変数名があれば置換後の変数名を変数配列6に登録する。ここで、変数配列6において、外部参照変数の場合には、変数名の後ろに、参照先変数名、参照先工程及びモジュール名が記述されていて、具体的な数値が入っていないので、このステップS4の時点では、属性及び変数名のみが読み込まれ、値は空白である(図11(A)及び図12(A)参照)。ここで、プログラム作成部2の変数配列作成部2eは、変数テーブル20も同様に読み込む。上述したように、変数テーブル20では、“(宣言された登録エリア)+1”個の登録エリアを持つ変数テーブル20に展開されて変数配列6に登録される。

[0078] なお、プログラム作成部2の変数配列作成部2eは、外部参照変数が記述されている要素定義ファイルを読み込んだ場合には、外部参照変数名、参照先変数名、参照先工程名及びモジュール名を一行ずつ読み出し、図13に示すような外部参照変数対応配列7を作成する(ステップS5)。この配列を作成するときも、要素配置データ3内に、変数名に関する変数名置換データがあり、かつ、この変数名置換データに、参照先名称に関して置換すべき変数名があれば、置換後の参照先名称を外部参照変数対応配列7に登録する。例えば、構成要素“140T・接続台・接続台”では、要素定義ファイル4cに記述されている外部参照変数において、“前搬送”が“コンベア”に置換され、“後搬送”が“搬送”に置換されて登録される。

[0079] 次に、要素定義ファイル4に初期処理プログラムが記述されていれば、プログラム作成部2の初期処理プログラム配列作成部2cにより、その初期処理プログラムが読み出されて図14に示すようば初期処理プログラム配列8に登録される(ステップS6)。同様に、プログラム作成部2のシミュレーションプログラム配列作成部2dにより、要素定

義ファイル4に記述されているシミュレーションプログラムが読み出されて図14に示すようなシミュレーションプログラム配列9に登録される(ステップS7)。なお、これらのプログラムの読み込みの際には、シミュレーションプログラムで使用する変数が、変数配列6内における当該変数の変数位置(アドレス)に対応付けられる。

[0080] 以上のステップS1〜S7の処理が、構成要素データ3を1行読み込んで、それに対応する1つの要素定義ファイル4を読み込む度に行なわれる。そして、この処理が構成要素データ3(要素配列5)の末行まで行なわれると、構成要素データ3に含まれる全ての構成要素に関してそれに関連するデータの読み込み及び配列の作成が完了する(ステップS8)。

[0081] ただし、この時点では、外部参照変数と参照先変数との対応がとれていないので(図11(A))、プログラム作成部2の変数対応部2fにより、その対応付けの処理を外部参照変数対応配列7を参照しながら行ない、変数配列6において空白となっている外部参照変数の値/位置の欄に、対応する取出変数の変数配列6内の変数位置(アドレス)を書き込む(ステップS9)。例えば、図11(A)において“140T・1ST・ユニット1”の外部参照変数“自工程加工準備完了”には、同一工程のジグ1の取出変数“加工準備完了”が対応しているので、外部参照変数“自工程加工準備完了”には、アドレス“R5”が登録される(図11(B)参照)。

[0082] ここで、取出変数として変数テーブル20が登録されている場合には、その変数テーブル20に対応する外部参照変数の値/位置欄には、変数配列6内での変数テーブル20の位置が順に登録されていく。図12(A)に示すように、上述した“機械状態T”という変数テーブル20の各テーブル変数には、各機械の外部参照変数“保全指令・機械状態”が対応しているので、変数配列6の作成当初には空白であった各機械の“保全指令・機械状態”の値/位置欄、例えば“保全指令・機械状態”という変数を持つ要素配置データ3で6つ目の要素(“140T・1ST・ユニット1”)の変数“保全指令・機械状態”には、変数テーブル“作業状態T”の6番目の取出変数の変数位置(ここでは“T0+6”)が登録される(図12(B)参照)。

[0083] 次に、プログラム実行部10により、初期処理プログラム配列8に含まれる初期処理プログラムが一命令ずつ実行される(ステップS10)。ここで例えば、初期処理プログ

ラムとして、“SET TBNO”命令が実行されると、そこで指定された変数の変数配列6内での位置データを、指定された外部参照変数の示すテーブル変数の値／位置欄に書き込み、同時に、そのテーブル変数の先頭テーブル変数“作業状態T[0]”のデータとしての登録個数を“1”加算する、という処理を行う。

[0084] 具体的には例えば、上述した構成要素“140T・1ST・ユニット1”に対応する初期処理プログラムにおいて、次のように記述されているものとする。

“SET TBNO “機械状態”データ位置 保全指令・機械状態”

この場合、変数配列6内での変数“機械状態”の位置データ(“Q6”)が“保全指令・機械状態”の示すアドレス“T0+6”の変数“機械状態T[6]”の値欄に書き込まれる(図12(C))。同様の処理が、“アंकランプT”という変数テーブル20と2つの“ジグ1”の変数“アंकランプ”との間で行なわれる(図9参照)。なお、“SET TBNO”命令は、変数テーブル20に対応する変数位置とテーブルへの登録数とを登録する登録手段を構成する。

[0085] これにより、図9及び図10に示すように、変数配列6には、シミュレーションで用いる全ての内部変数、外部参照変数及び取出変数が含まれ、かつ、外部参照変数と取出変数とが対応付けられ、かつ、まとめて管理すると好適な変数とその変数をまとめて管理する変数テーブル20とが対応付けられるので、全ての変数の値は、この変数配列6を参照することにより求められる。

[0086] 次いで、入力装置11からシミュレーションの指示があると(ステップS11)、プログラム実行部10により、シミュレーションプログラム配列9に含まれる要素配列5に従って並べられた一連のシミュレーションプログラムが先頭から順番に一命令ずつ実行される(ステップS12)。ここで、シミュレーションプログラムは、各命令の記述が文字情報であるので、その命令を読む都度、プログラム実行部10が文字情報による命令記述を解釈して実行してもよいが、予め各命令をそれらの各命令に対応して設定された数値に置き換えておき、その命令に対応する数値を読み出して、その数値に対応した命令処理をプログラム実行部10が実行するようにすると、実行速度が上がって好ましい。

[0087] このようにしてシミュレーションプログラムが実行されると、変数配列6の対応する変

数の値が刻々と変化する。

[0088] シミュレーションプログラムは先頭から末尾まで一定のスキャンピッチ(シミュレーションを行なう上での経過時間)(例えば0.1秒間隔)で繰り返し実行される。実行にあたって、変数配列6の各変数が参照される。このスキャンピッチは任意に設定することができ、例えば、実際の機械の動作時間(実時間)でシミュレーションするように設定することもできるし、逆に、短くして早送りのようにシミュレーションすることも可能である。外部参照変数では、参照先変数が、その外部参照変数の示すアドレス(変数配列6内の位置)を頼りに読み出される。工程シミュレーションを行なう上で所定の時間が経過した時点で(あるいは1スキャンが終わるごとに)、実行を停止することができる。

[0089] このとき、入力装置11により、特定の構成要素を指定してシミュレーション結果の表示を指示すれば、要素配列5に基づいて変数配列6の変数格納範囲が読み出され、出力装置12により、工程識別名称及び構成要素名称と共に変数名とその値とが表示され、これにより、シミュレーション結果を確認することができる(ステップS13)。その際、構成要素を表示画面上で模式的に表示すると共に、変数の値の変化を構成要素の動作としてアニメーション表示させるようにすると、更に分かりやすい。

[0090] なお、シミュレーションプログラムを実行するとき、構成要素“保全指令1”では、“機械状態T”の変数テーブルを用いて全ての機械状態(異常なしか、故障中か)を判断し、“作業員状態T”テーブルを用いて、空いている保全メンバを検索して、空いている保全メンバを要修理機械の修理に向かわせる、といったシミュレーションが行なわれる。このようなシミュレーションにおいては、変数テーブル20として“機械状態T”テーブルを持たなくても、変数配列6を先頭から末尾まで順に検索すれば、全ての“機械状態”データを知ることができるが、上述したようにして、構成要素ごとの“機械状態”データについて、その変数位置(変数配列6内でのアドレス)をまとめて“機械状態T”テーブルに集約しておければ、その限られた記憶領域のデータについて参照することで全ての“機械状態”データを知ることができるので、シミュレーションプログラムの実行時間を短くすることができる。

[0091] 以上のようにして、ある特定の要素配置データ3に関する工程シミュレーションを行なうことができる。しかしながら、例えば、“140T・2ST”の後に“140・2ST”と同一の

工程を増やして更に工程シミュレーションを行ないたい場合には、要素配置データ3に、表計算ソフトの編集機能を利用して“140T・2ST”の1行をコピーして“140T・2ST”の下に1行追加してこれを“140T・3ST”とし、更に、この加工工程(“ジグ1”、“ユニット1”)の追加により変更しなければならない関連するデータ(例えば、“アンプT”テーブルのデータ登録エリア数、“機械状態T”テーブルのデータ登録エリア数等)を修正するとよい。この状態で、図7に示すフローチャートに従って、図1に示す工程シミュレーションシステム1を動作させてやれば、追加した“ジグ1”と“ユニット1”の間での変数間の対応については、要素配置データ3上での配置により、“ジグ1”の要素定義ファイル4f、“ユニット1”の要素定義ファイル4gに記述されている相対的な参照先の指示により一義的に定まるので、要素定義ファイル4f、4gに変更がない限り、新たに追加した工程が反映されたシミュレーションプログラム配列9、変数配列6及びその他の配列が作成される。そして、このようにして作られたシミュレーションプログラム配列9に含まれる一連のシミュレーションプログラムを実行すれば、新たな工程編成による工程シミュレーションを容易に行なうことができる。

請求の範囲

- [1] 生産システムの工程シミュレーションを行なう工程シミュレーションシステムにおいて、
- 生産システムを構成する各種の構成要素の配置を工程識別名称と構成要素名称との組み合わせによりワークの流れに沿って記述してなる要素配置データから、構成要素名称を工程識別名称と組み合わせた状態で順次読み込む要素配置データ読込手段と、
- 前記各種の構成要素の動作シミュレーションを行なうためのシミュレーションプログラムを前記各種の構成要素ごとに記述してなる複数の要素定義ファイルの中から、前記要素配置データ読込手段により読み込まれた構成要素名称に対応する要素定義ファイルを順次読み込む要素定義ファイル読込手段と、
- 前記要素定義ファイル読込手段により読み込まれた前記各要素定義ファイルに記述されているシミュレーションプログラムを順に配列してシミュレーションプログラム配列を作成するプログラム配列作成手段と、
- 前記プログラム配列作成手段により作成された前記シミュレーションプログラム配列に含まれる一連のシミュレーションプログラムを実行して、前記生産システムを構成する前記各種の構成要素の動作をシミュレートするプログラム実行手段とを備えたことを特徴とする工程シミュレーションシステム。
- [2] 前記要素配置データは、文字情報の追加や削除等の編集が可能なソフトウェアを用いて作成される表形式のデータからなり、列方向にワークの流れ方向が設定され、行方向に工程識別名称と構成要素名称とが記述されていることを特徴とする、請求項1に記載の工程シミュレーションシステム。
- [3] 前記各要素定義ファイルは、自己のシミュレーションプログラムを記述したプログラム記述部と、自己のシミュレーションプログラムで使用される変数を記述した変数記述部とを含み、前記変数記述部には、他の要素定義ファイルの変数を参照する要素定義ファイルであれば、参照先変数を指定する外部参照変数が定義され、他の要素定義ファイルに変数を参照させる要素定義ファイルであれば、外部参照変数により参照される取出変数が定義されており、

前記要素定義ファイル読込手段により読み込まれた前記各要素定義ファイルの前記変数記述部に記述されている全ての変数を含む変数配列を作成する変数配列作成手段と、

前記変数配列作成手段により作成された前記変数配列に含まれる前記外部参照変数と前記取出変数とを対応付ける変数対応手段とをさらに備えたことを特徴とする、請求項2に記載の工程シミュレーションシステム。

- [4] 前記要素配置データには、前記各要素定義ファイルの前記変数記述部に記述されている変数名を別の変数名に置換するための変数名置換データが記述されており、

前記要素配置データにおいて前記変数名置換データが設定されている変数名を前記変数名置換データに記述されている別の変数名に置換する変数名置換手段をさらに備えたことを特徴とする、請求項3に記載の工程シミュレーションシステム。

- [5] 前記変数配列作成手段により作成された前記変数配列には、複数の構成要素間に亘る複数の同種の変数をまとめて管理するための登録エリアを所要数備えた変数テーブルが作成されており、

前記各要素定義ファイルのうち、まとめて管理される同種の変数を含んでいる要素定義ファイルには、当該変数に応じて作成される前記変数配列内の対応する変数テーブルに、前記変数配列内における当該変数の変数位置を登録する初期処理プログラムが記述されており、

前記プログラム配列作成手段は、前記要素定義ファイル読込手段により読み込まれた前記各要素定義ファイルに記述されている初期処理プログラムを順に配列して初期処理プログラム配列を作成し、

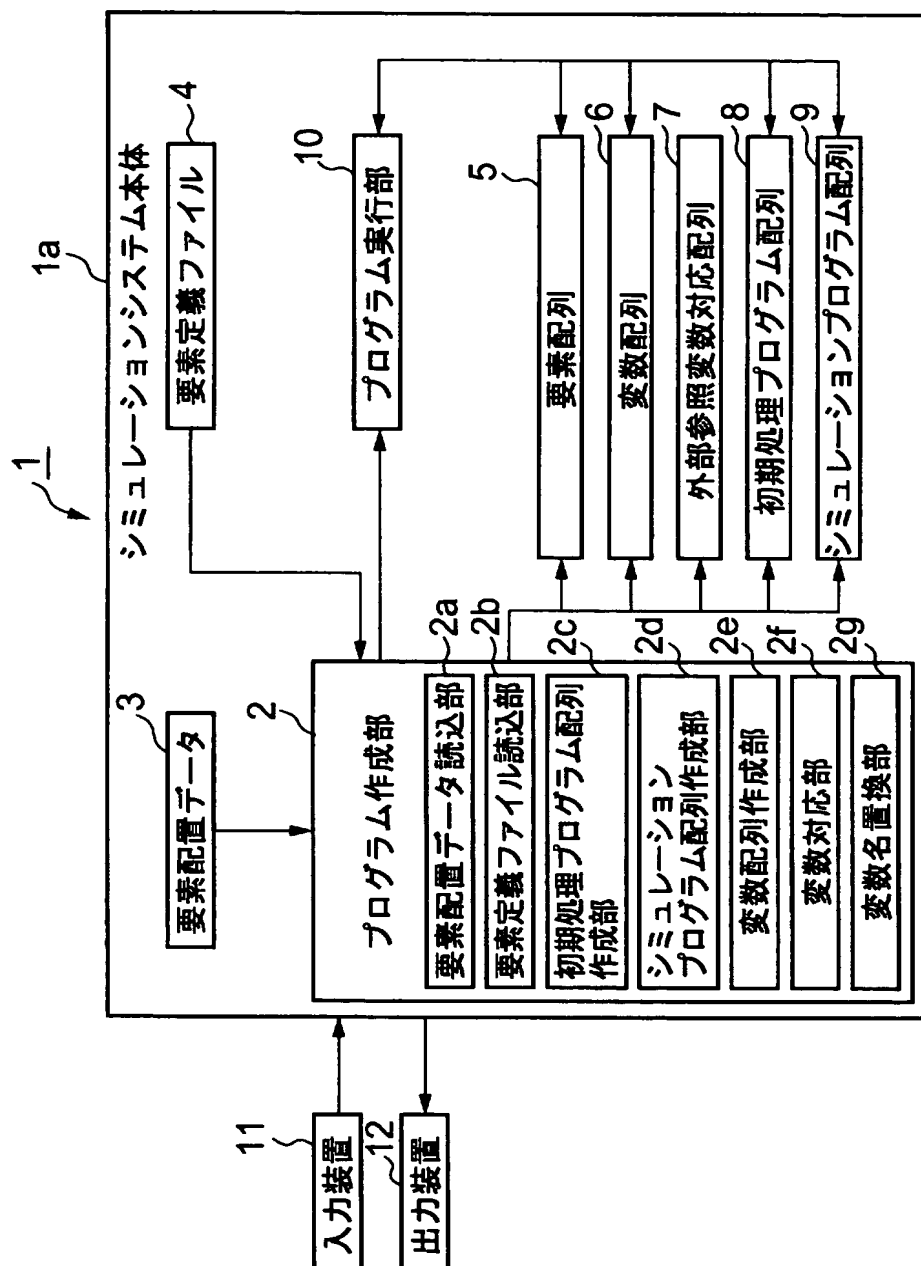
前記プログラム実行部は、前記プログラム配列作成手段により作成された前記初期処理プログラム配列に含まれる初期処理プログラムを実行して、まとめて管理される同種の変数について、前記変数配列内の変数テーブルに、前記変数配列内における当該変数の変数位置を登録することを特徴とする、請求項3又は4に記載の工程シミュレーションシステム。

- [6] 前記各要素定義ファイルの前記プログラム記述部に記述されているシミュレーショ

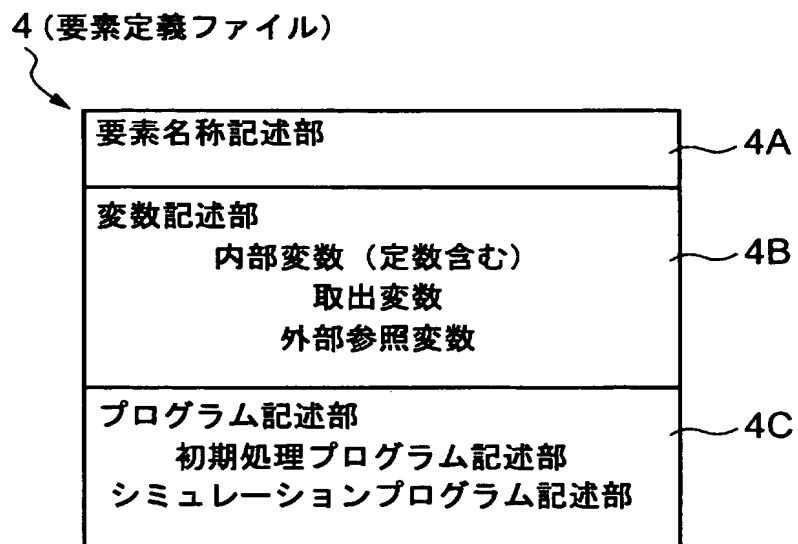
ンプログラムは、ラダー言語方式の命令群で記述されていることを特徴とする、請求項1〜5のいずれか一項に記載の工程シミュレーションシステム。

- [7] 請求項1〜6のいずれか一項に記載の工程シミュレーションシステムとしてコンピュータを機能させるための工程シミュレーションシステム用プログラムを記録したコンピュータ読み取り可能な記録媒体。

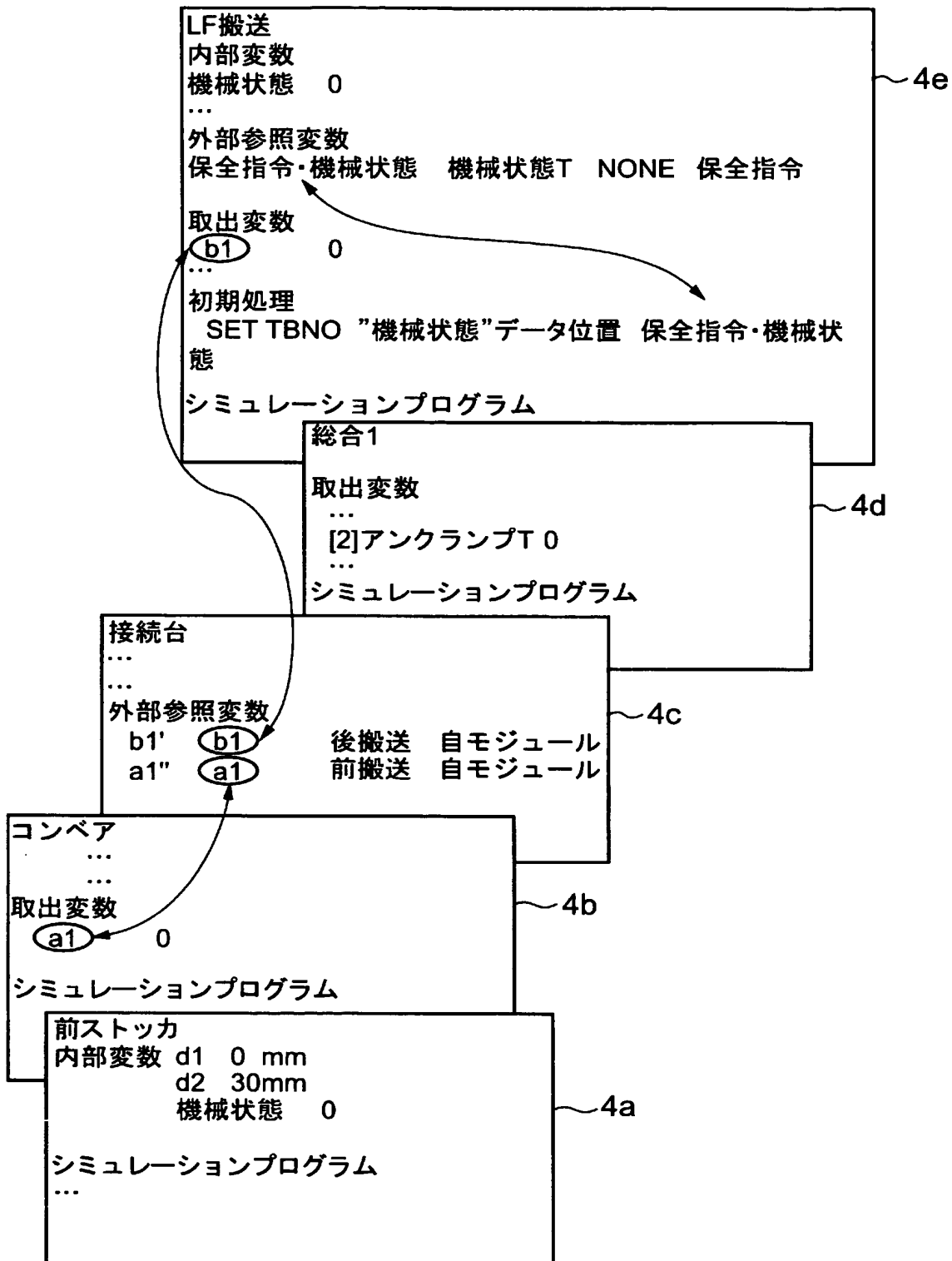
[図1]



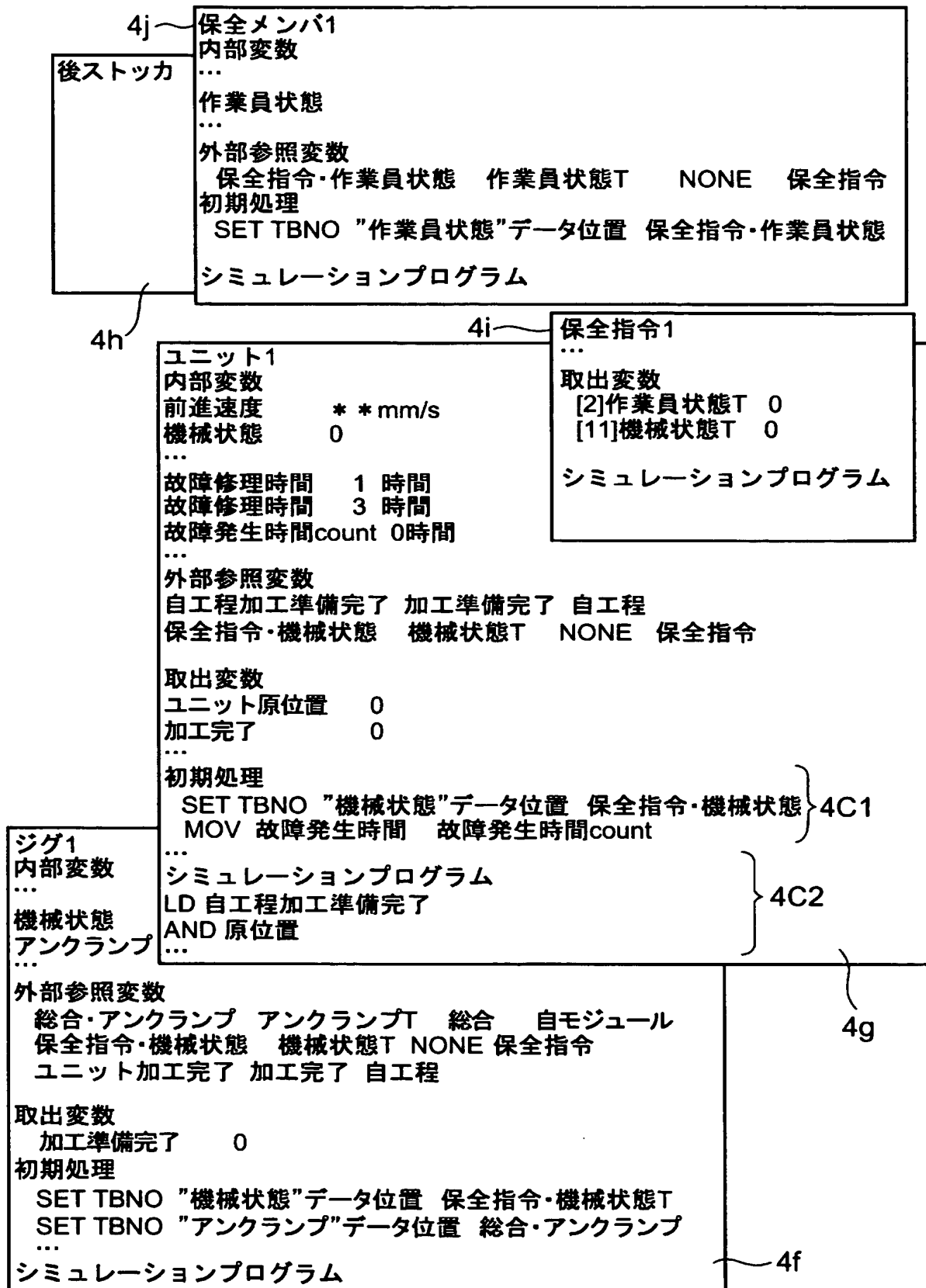
[図3]



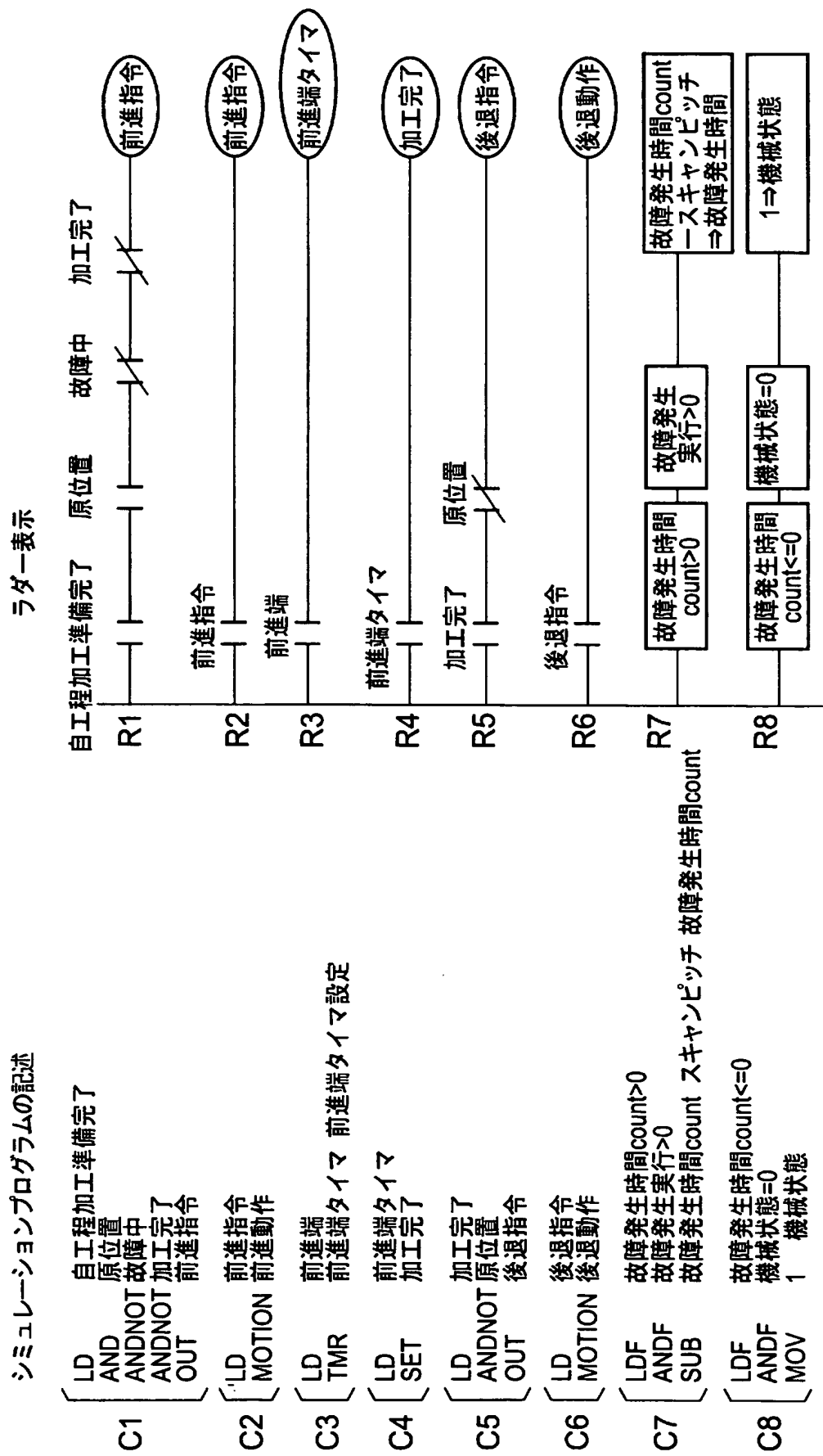
[図4]



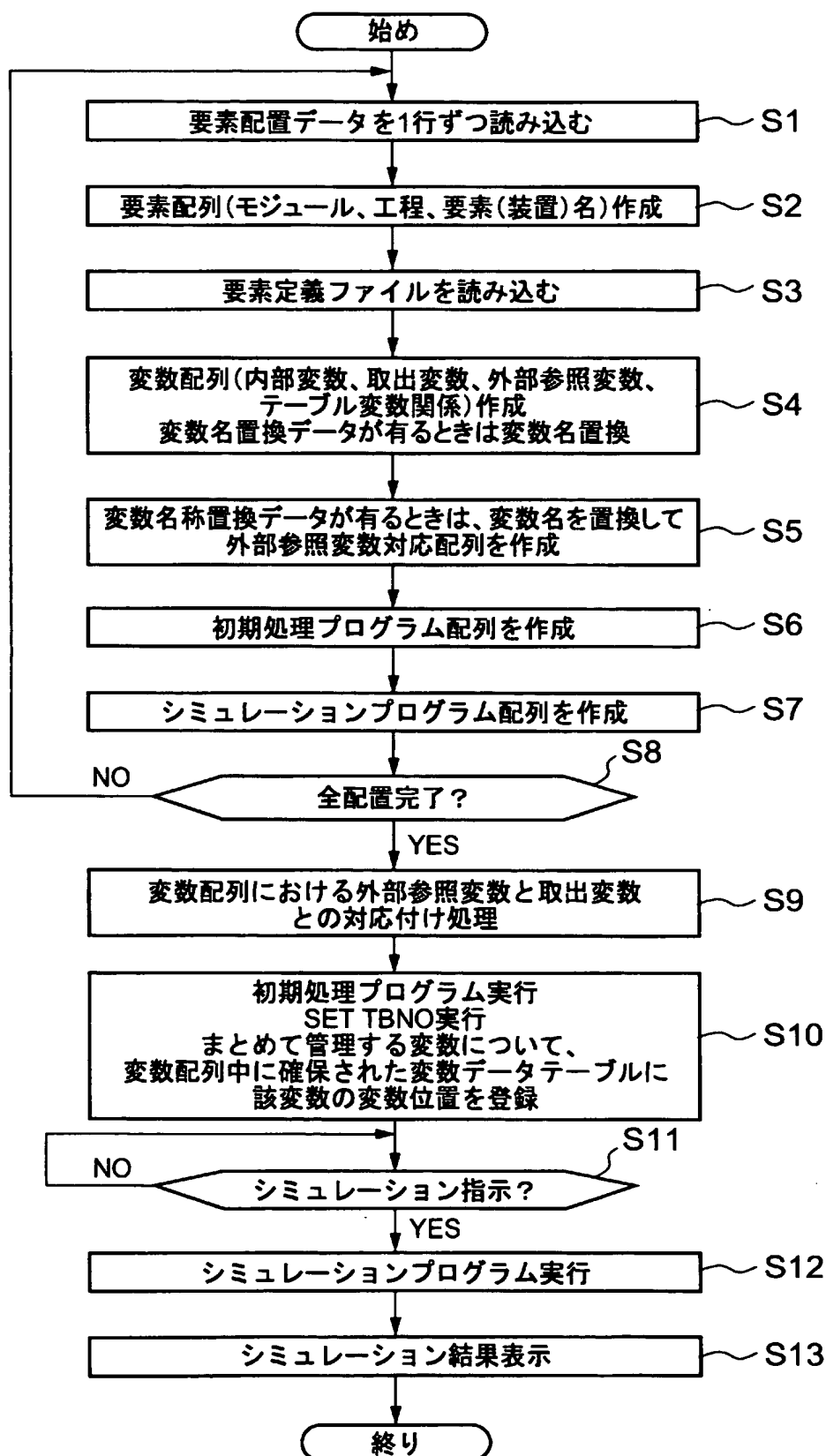
[図5]



[図6]



[図7]



[図8]

5 (要素配列)

工程識別名称

要素配列内位置	モジュール名	工程名	要素名	変数先頭位置	個数
1	前ストッカ		前ストッカ	1	PO
2	140T	コンベア	コンベア	n1+1	P1
3	140T	接続台	接続台	n2+1	P2
4	140T	総合	総合1	n3+1	...
5	140T	搬送	LF搬送
6	140T	1ST	ジグ1
7	140T	1ST	ユニット1		
8	140T	2ST	ジグ1		
9	140T	2ST	ユニット1		
10	140T	後接続台	接続台		
11	140T	搬出コンベア	コンベア		
12	後ストッカ		後ストッカ		
13	保全指令		保全指令1		
14	保全メンバA		保全メンバ
15	保全メンバA		保全メンバ

[図9]

変数位置 (アドレス)	属性	6 (変数配列) ↓ 変数名	値/位置	
1	内部	d1	0mm	前ストッカ
2	内部	d2	30mm	
		...		
Q1	内部	機械状態	0	
		...		
n1+1		...		コンベア(140Tコンベア)
R3	取出	a1	* * *	
		...		
n2+1		...		接続台(140T接続台)
	外部参照	b1'	R4	
	外部参照	a1''	R3	
		...		
n3+1				総合1(140T総合) アंकランプTテーブル 20
L0	取出	アंकランプT[0]	2	
L0+1	取出	アंकランプT[1]	f5	
L0+2	取出	アंकランプT[2]	...	
n4+1		...		LF搬送(140T搬送)
R4	取出	b1	* *	
		...		
		...		
n5+1	内部	...		ジグ1(140T1ST)
Q5	内部	機械状態	0	
f5	内部	アंकランプ	0	
...	外部参照	総合・アंकランプ	L0+1	
...	外部参照	保全指令・機械状態	T0+5	
...	外部参照	ユニット加工完了	R7	
R5	取出	加工準備完了	0	
		...		
n6+1	内部	前進速度	* mm/s	ユニット1(140T1ST)
Q6	内部	機械状態	0	
		...		
	内部	故障発生時間	1時間	
	内部	故障修理時間	3時間	
	内部	故障発生時間count	0時間	
	内部	故障発生実行	1 (有り)	
		...		
	取出	ユニット原位置	* *	

(次図へ続く)

[図10]

(前図からの続き)

変数位置 (アドレス)	属性	変数名	値/位置
R7	取出	加工完了	0
		...	
R6	外部参照	自工程加工準備完了	R5
t6	外部参照	保全指令・機械状態	T0+6
		...	
		...	
T0	取出	機械状態T	11
T0+1	取出	機械状態T[1]	Q1
T0+2	取出	機械状態T[2]	Q2
...	取出	...	
T0+11	取出	機械状態T[11]	Q11
“ 作業員状態T ” テーブル			
		...	

6 (変数配列)

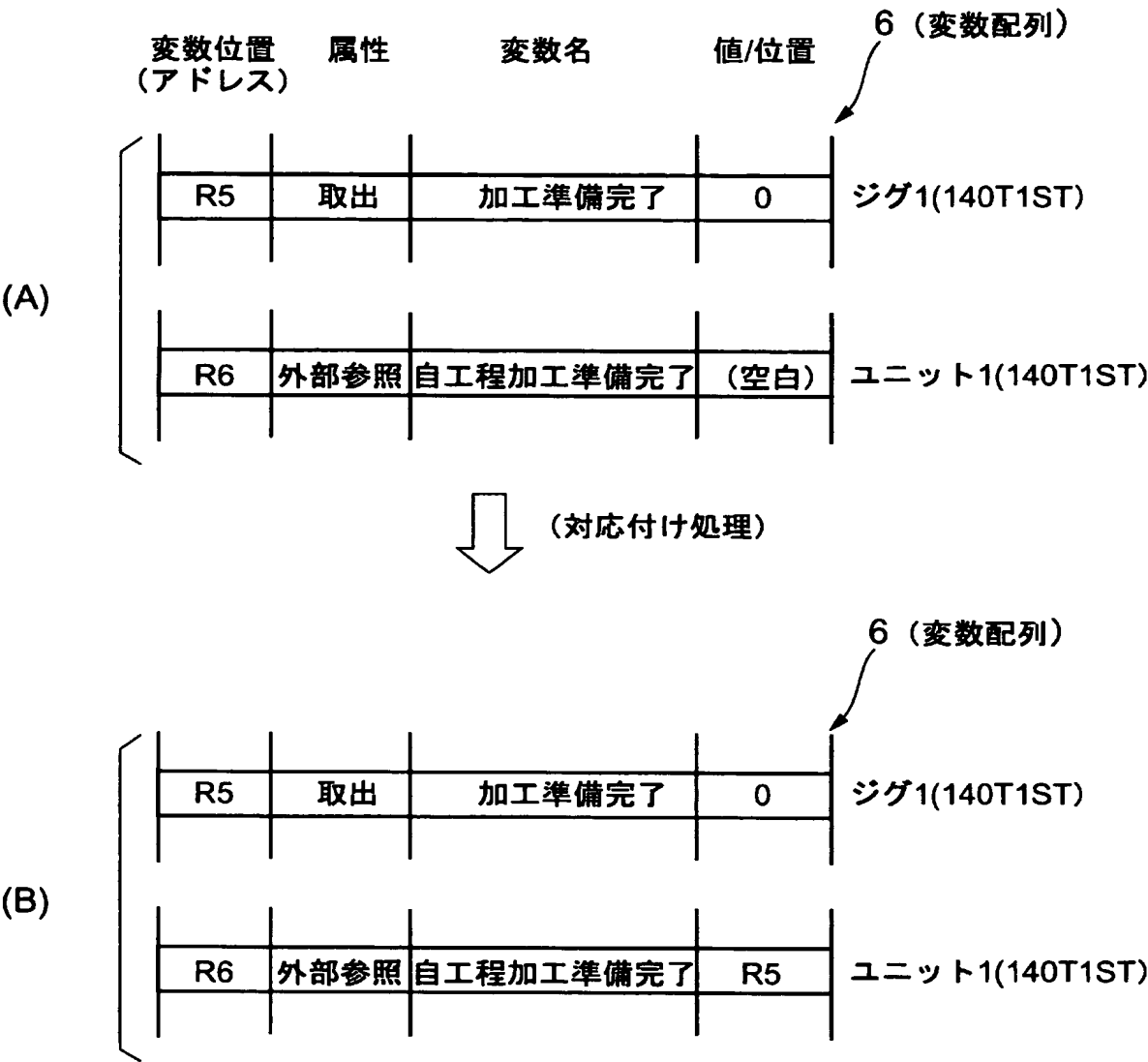
ユニット1(続き)

ジグ1(140T2ST)
ユニット1(140T2ST)
接続台:140T後接続台
コンベア(140T搬出コンベア)
後ストッカ (後ストッカ)

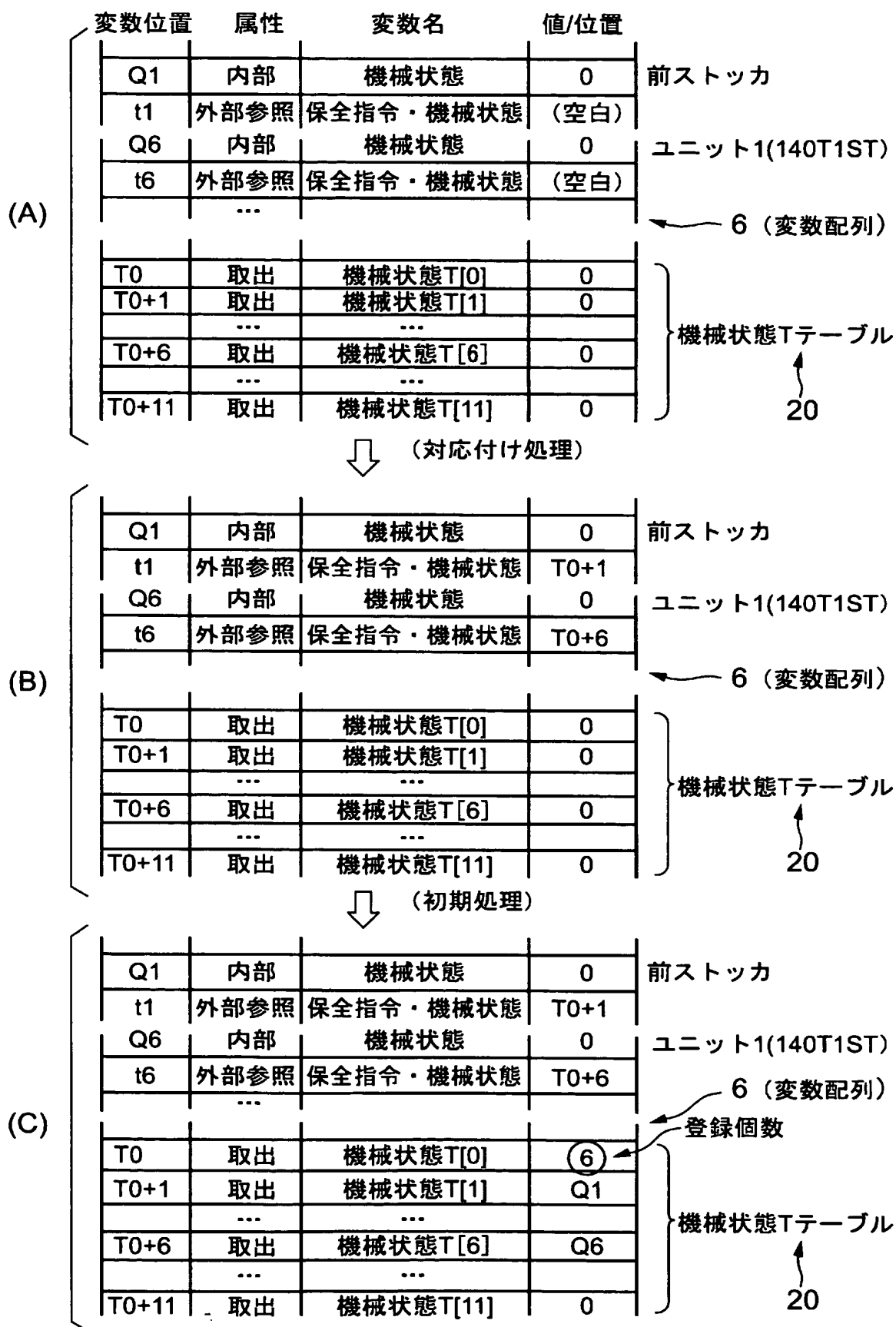
保全指令1(保全指令)
機械状態Tテーブル
20

20
保全メンバ(保全メンバA)
保全メンバ(保全メンバB)

[図11]



[図12]



[図13]

7 (外部参照変数対応配列)

要素配列内位置	外部参照変数名	参照先変数名	参照先モジュール名工程名
3	b1'	b1	140T搬送
3	a1''	a1	140Tコンペア
...		...	
...		...	
7	自工程加工準備完了	加工準備完了	140T1ST
...			


[図14]

8 (初期処理プログラム配列)

前ストッカ
コンペア
接続台
総合1
LF搬送
ジグ1
ユニット1
ジグ1
ユニット1
接続台
コンペア
後ストッカ
保全指令
保全メンバ
保全メンバ

[図15]

9 (シミュレーション
プログラム配列)

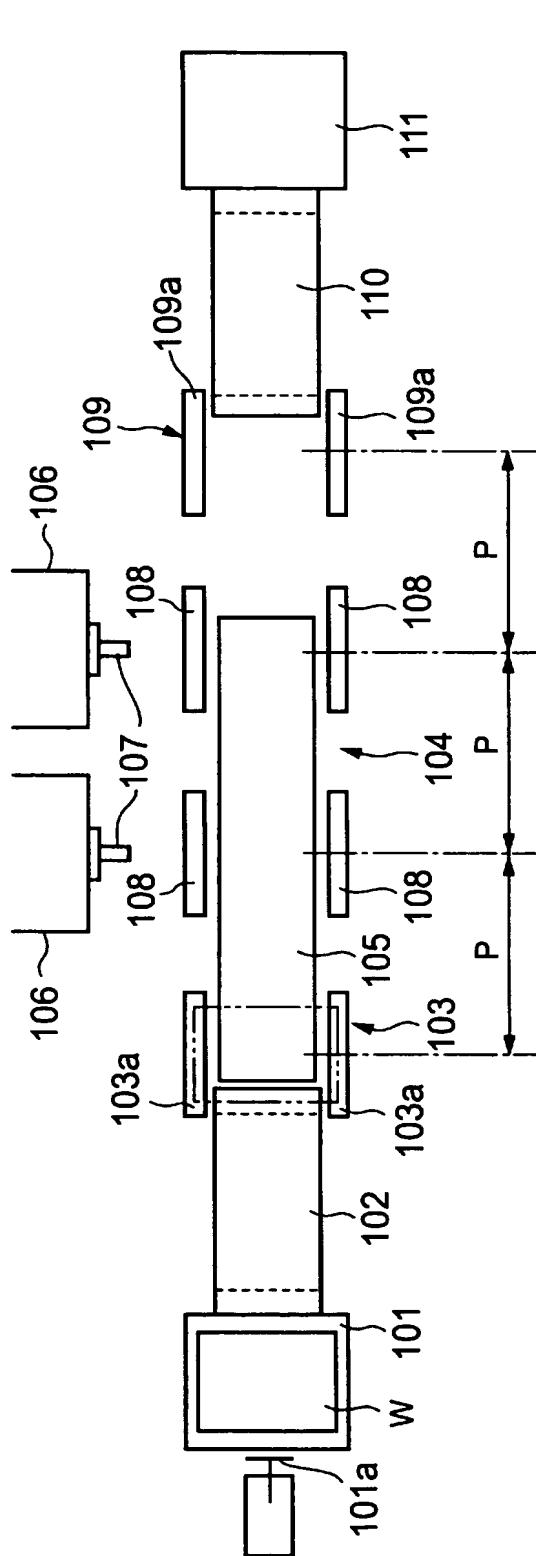


前ストッカ
コンペア
接続台
総合1
LF搬送
ジグ1
ユニット1
ジグ1
ユニット1
接続台
コンペア
後ストッカ
保全指令
保全メンバ
保全メンバ

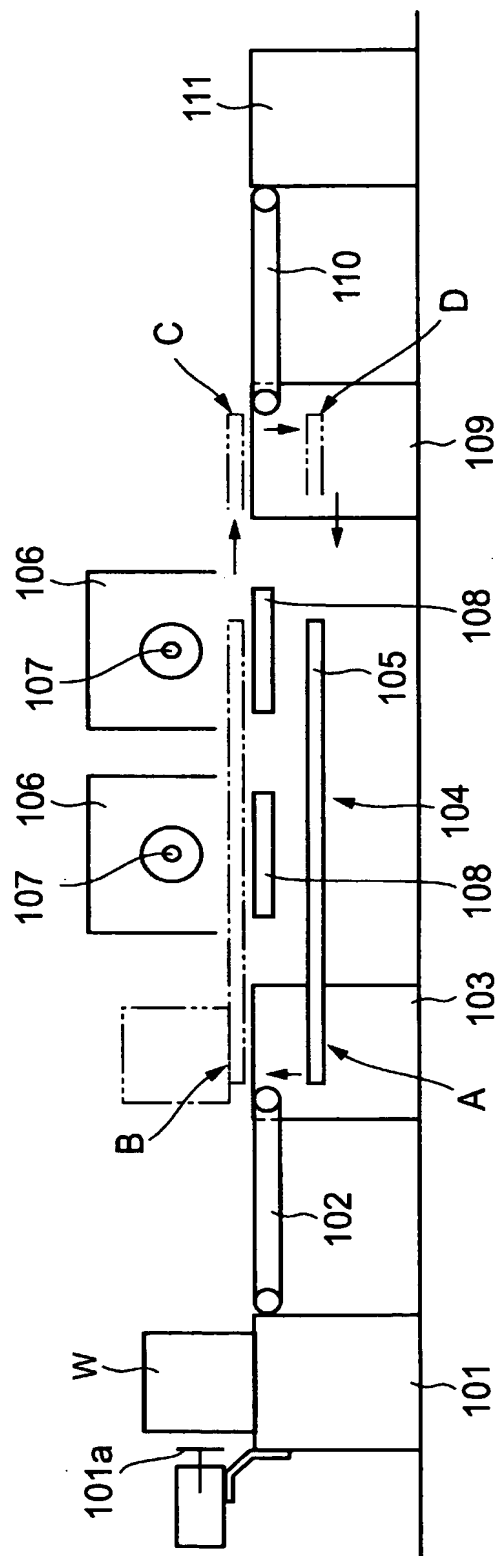
[図16]

モジュール名	工程名	変数名置換データ
140T	1ST	ユニット1 {Lユニット原位置=Lユニット=ユニット1}
140T	1ST	ユニット1 {Rユニット原位置=Rユニット=ユニット1}

[図17]



17A



17B